An Analysis of Initial Training Strategies for Exemplar-Free Class-Incremental Learning

Grégoire Petit*[1,2], Michael Soumm*[1], Eva Feillet*[1,4], Adrian Popescu[1],
Bertrand Delezoide[3], David Picard[2], Céline Hudelot[4]
[1]Université Paris-Saclay, CEA, LIST, F-91120, Palaiseau, France
[2]LIGM, Ecole des Ponts, Univ Gustave Eiffel, CNRS, Marne-la-Vallée, France
[3]Amanda, 34 Avenue Des Champs Elysées, F-75008, Paris, France
[4]Université Paris-Saclay, CentraleSupélec, MICS, France
g.petit360@gmail.com, {michael.soumm, eva.feillet, adrian.popescu}@cea.fr,
david.picard@enpc.fr,bertrand.delezoide@amanda.com,celine.hudelot@centralesupelec.fr

Abstract

Class-Incremental Learning (CIL) aims to build classification models from data streams. At each step of the CIL process, new classes must be integrated into the model. Due to catastrophic forgetting, CIL is particularly challenging when examples from past classes cannot be stored, the case on which we focus here. To date, most approaches are based exclusively on the target dataset of the CIL process. However, the use of models pre-trained in a self-supervised way on large amounts of data has recently gained momentum. The initial model of the CIL process may only use the first batch of the target dataset, or also use pre-trained weights obtained on an auxiliary dataset. The choice between these two initial learning strategies can significantly influence the performance of the incremental learning model, but has not yet been studied in depth. Performance is also influenced by the choice of the CIL algorithm, the neural architecture, the nature of the target task, the distribution of classes in the stream and the number of examples available for learning. We conduct a comprehensive experimental study to assess the roles of these factors. We present a statistical analysis framework that quantifies the relative contribution of each factor to incremental performance. Our main finding is that the initial training strategy is the dominant factor influencing the average incremental accuracy, but that the choice of CIL algorithm is more important in preventing forgetting. Based on this analysis, we propose practical recommendations for choosing the right initial training strategy for a given incremental learning use case. These recommendations are intended to facilitate the practical deployment of incremental learning.

*denotes equal contribution

Initial training strategy					CIL Algorithms					
					BSIL [30]		DSLDA [22]		FeTrIL [47]	
Arch	Method	FT	Ext	Sup	$\mu_{\overline{Acc}}$	W	$\mu_{\overline{Acc}}$	W	$\mu_{\overline{Acc}}$	W
RN50	CE	✓	×	SL	44.9	0	53.7	4	51.0	0
RN50	CE	×	✓	SL	39.9	0	61.4	0	60.6	0
RN50	CE	✓	✓	SL	62.9	1	65.3	0	68.4	1
RN50	BYOL	✓	×	SSL	11.2	0	42.2	0	34.4	0
RN50	BYOL	×	✓	SSL	35.3	0	63.3	0	62.0	0
RN50	BYOL	✓	✓	SSL	60.2	0	70.0	2	70.2	0
RN50	MoCoV3	✓	×	SSL	14.9	0	49.6	0	41.1	0
RN50	MoCoV3	×	✓	SSL	36.3	0	67.9	1	65.3	0
RN50	MoCoV3	✓	✓	SSL	64.7	2	71.8	2	72.0	0
ViT-S	DeiT	×	✓	SL	35.0	0	58.7	0	56.3	0
ViT-S	DeiT	✓	✓	SL	11.2	0	37.4	0	27.4	0
ViT-S	DINOv2	×	✓	SSL	70.4	4	75.7	9	72.4	6
ViT-S	DINOv2	✓	✓	SSL	24.0	0	45.9	0	39.2	0

Table 1. Performance of three EFCIL algorithms with different training strategies for the initial model, averaged over 16 target datasets and two EFCIL scenarios. BSIL [30] is a recent EFCIL algorithm which is representative of fine-tuning-based CIL works. DSLDA [22] and FetrIL [47] adapt linear probing [32] for EFCIL. We present the averaged incremental accuracy ($\mu_{\overline{Acc}}$) and the number of cases (W) in which a combination of algorithm and initial training strategy performs best for a combination of target dataset and EFCIL scenario (see Sec.4). Initial training strategies are defined by: Arch- deep architecture used (ResNet50 (RN50) [25] or vision transformer (ViT-S) [11]); Method - initial training method; FT - fine-tuning on initial classes of the target dataset; Ext- use of an external dataset, such as ILSVRC [52]; Sup - type of supervision for the initial model: self-supervised (SSL) or supervised (SL).

1. Introduction

Real-world applications of Machine Learning (ML) often involve training models from data streams characterized by distributional changes and limited access to past data [23,59]. This scenario presents a challenge for standard ML algorithms, as they assume that all training data is avail-

able at once. Continual learning addresses this challenge by building models designed to incorporate new data while preserving previous knowledge [51]. Class-Incremental Learning (CIL) is a type of continual learning that handles the case where the data stream is composed of batches of classes. It is particularly challenging in the exemplar-free case (EFCIL), i.e. when storing examples of previous classes is impossible due to memory or confidentiality constraints [22, 71]. CIL algorithms must find a balance between knowledge retention, i.e. stability, and adaptation to new information, i.e. plasticity [38, 40, 66]. Many existing EFCIL methods [30, 34, 35, 49, 69–71] update the model at each incremental step using supervised fine-tuning combined with a distillation loss, and thus tend to favor plasticity over stability. Another line of work [22, 47] freezes the initial model and only updates the classifier. This approach has recently gained interest [28, 46, 64] due to the availability of models pre-trained on large external datasets, often through self-supervision [24, 43]. While pre-trained models provide diverse and generic features, there are limits to their transferability [1], and these limits have not been studied in depth in the context of EFCIL.

We propose a comprehensive analysis framework to disentangle the factors which influence EFCIL performance. Focus is put on the strategies to obtain the initial model of the incremental process. We consider the type of neural architecture, the training method, the depth of fine-tuning, the availability of external data, and the supervision mode for obtaining this initial model. The initial training strategies are compared using three EFCIL algorithms, representative for the state of the art, on 16 target datasets, under 2 challenging CIL scenarios. The obtained results are summarized in Table 1. The main findings are that: (1) pre-training with external data improves accuracy, (2) self-supervision in the initial step boosts incremental learning, particularly when the pre-trained model is fine-tuned on the initial classes, and (3) EFCIL algorithms based on transfer learning have better performance than their fine-tuning-based counterparts. However, the distribution of best performance, presented in Table 1, shows that no combination of an EFCIL algorithm and an initial training strategy is best in all cases. This echoes the results of previous studies such as [6, 14]. Therefore, it is interesting to understand the contribution of the different factors influencing EFCIL performance. To this aim, we analyze these strategies in depth in Section 5, and use this analysis to formulate EFCIL-related recommendations in Section 6. The insights brought by the proposed analysis could benefit both continual learning researchers and practitioners. The proposed framework can improve the evaluation and analysis of EFCIL methods. Continual learning practitioners can use the results of this study to better design their incremental learning systems.

2. Background

2.1. Pre-training methods

Transfer learning involves using a model trained on a source dataset as a starting point for training another model on a target dataset [50]. In the case of transfer learning, the weights of the target model are generally initialized with the weights of the source model. These weights can remain fixed, except for the classification layer (*linear probing*), or they can be updated using the target data (*fine-tuning*). Transfer learning has several practical advantages [55]. It reduces the computational effort to train a new model on a new dataset. It also enables learning an accurate model in few-shot settings, because models pre-trained on large datasets are able to extract complex features even for new input data. Some authors investigate how to pre-train the model in order to make it more transferable [19, 31, 54]. Model generalization is favored by the quantity, quality, and diversity of its source training data [43]. However, the parametric footprint of pre-trained models, typically in the range of hundreds of millions, is often too high for continual learning applications [23]. Smaller models can be obtained from larger models through knowledge distillation [26, 58].

Self-Supervised Learning (SSL) has recently gained interest thanks to its ability to produce diverse, reusable features for downstream tasks. SSL enables a model to learn from unlabeled data without relying on explicit annotations [29]. It leverages the inherent structure or information present within the data itself to create surrogate labeling tasks e.g. predicting missing image patches, image rotations, or colorizations. For example, MoCov3 [9, 24] uses a contrastive loss function to obtain similar representations for two randomly augmented crops of the same input image. Recently, SSL methods trained on large datasets such as BYOL [20] and DINOv2 [43] have provided efficient feature extractors, reusable for other tasks. We note that, while the reuse of pre-trained models as frozen feature extractors is easy, their fine-tuning in the presence of domain shift may be challenging [32]. This is important in the context of CIL since many existing models are based on fine-tuning.

We compare various pre-training methods to obtain the initial model of a CIL process (Subsec. 3.2). We consider (i) the case where the initial model is trained using only the initial batch of target data and (ii) the case where an external dataset was available for pre-training. In the first case, the initial model is either obtained using classic supervised learning or using an SSL algorithm, here MoCov3 [9]. In the second case, we start the EFCIL process with a model whose weights have been learned either in a supervised manner or in a self-supervised manner [20, 43] or through distillation [58]. This allows us to study the transferability of the resulting initial models, in combination with various CIL methods.

2.2. Class-Incremental Learning (CIL)

Continual learning aims to build models that are able to continuously and adaptively learn about their environment. In CIL, learning a classification model is a sequential process, where each step in the sequence consists of integrating a set of new classes into the model [6, 33, 38, 45]. In the exemplar-free setting, at a given stage in the process, the model must be able to recognize all the classes encountered so far, with access only to the current batch of classes.

The main challenge faced by CIL models is their tendency to forget previously acquired information when confronted with new information. This phenomenon is called *catastrophic forgetting* or *catastrophic interference*, as it is caused by the "interference" of new information with previous information [16, 39]. Forgetting may be reduced by storing examples from past classes, a strategy called *rehearsal* [49]. However, the availability of past data and the possibility to store it may be unrealistic in practice. Thus, we focus on EFCIL rather than rehearsal-based CIL.

Two main directions may be considered to deal with forgetting in artificial neural networks. A first family of CIL approaches lets the network grow as new capabilities must be learned, e.g. [62]. In the extreme case, this approach can result in zero forgetting. But at the same time, it is not realistic to make the model grow infinitely. A second family of methods considers a network of constant size throughout the incremental process (except the classifier) and proposes various strategies for obtaining models which ensure a balance between stability, i.e. preserving the performance for past classes, and plasticity, i.e. learning to recognize new classes. The weights of the initial model may be fine-tuned in combination with knowledge distillation between the previous model and the model which is currently learned [12, 27, 30, 34, 69–71]. This type of approach favors plasticity over stability because at each incremental step, all model weights are updated using the training images of the latest classes. An alternative is to use a pre-trained model or to freeze the model learned in the initial incremental state and to train only a linear classification layer afterward [5, 22, 47]. Recent works propose to use a large pre-trained model combined with a k-NN classifier as a challenging baseline for continual learning algorithms [28, 46]. These works adapt linear probing to an incremental context. They favor stability over plasticity since the feature extractor is not adapted during the incremental process [38]. The main challenge lies in the transferability of the initial feature extractor to new classes. Importantly, they are much faster to train than the fine-tuning-based methods because only the classification layer is updated.

Recent works propose to improve the learned representation by fine-tuning the model with a combination between cross-entropy loss and a self-supervised learning objective [15, 56]. Another recent trend in CIL is to use a pre-trained model as an efficient starting point for the incremental process [57, 67]. The authors of [17] explore the use of a fixed feature extractor pre-trained in a self-supervised way. The authors of [64] also propose a method based on dynamic prompting. In [44], pre-trained models are used to propose a compute-low method with a replay of past training samples. From a practical perspective, using a pre-trained feature extractor is also interesting for cases where training data is scarce, as in few-shot CIL [2].

The present work proposes a comprehensive study of training strategies for the initial model, with a focus on the interaction of these methods with different EFCIL algorithms. We experiment with transformer-based and CNN architectures, in combination with fine-tuning-based and transfer-learning-based EFCIL algorithms.

3. Problem statement

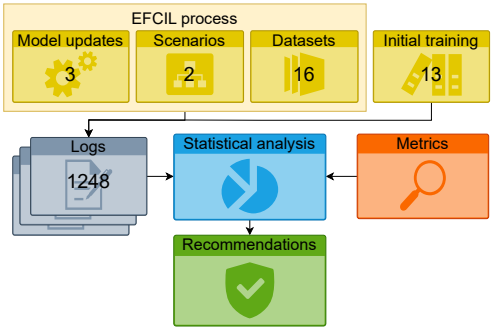

Figure 1. Overview of the proposed analysis framework of initial training strategies for EFCIL.

We summarize our proposed analysis framework in Figure 1. It combines a comprehensive modeling of the EFCIL process and initial training strategies as inputs for a statistical analysis that uses different EFCIL metrics. Recommendations for the design of EFCIL approaches are made based on the conclusions of the statistical analysis.

3.1. EFCIL process

Let us consider a dataset $\mathcal{D}$ split over K subsets, $\mathcal{D} = \mathcal{D}_1 \cup \mathcal{D}_2 \cup \cdots \cup \mathcal{D}_K$, and an EFCIL algorithm $Incr$. A CIL process consists in learning a classification model sequentially over K non-overlapping steps using $Incr$. At each step $k \in [\![1, K]\!]$, the model is updated using $Incr$ and the data subset $\mathcal{D}_k$, whose associated set of classes is denoted by $\mathcal{C}_k$. The data subsets $\mathcal{D}_1, \mathcal{D}_2, \cdots, \mathcal{D}_K$ composing the complete dataset $\mathcal{D}$ satisfy the following constraint: for $k, k' \in \{1, 2, \ldots, K\}$ with $k \neq k'$, $\mathcal{C}_k \cap \mathcal{C}_{k'} = \emptyset$, i.e. each class is only present in a single data subset. The use of an exemplar-free algorithm $Incr$ implies that when the training is performed at the k^{th} step, no example from any of the data subsets of the previous steps can be accessed. Although this is a more difficult setting, it is also more realistic

in practice [6, 23].

Incremental model updates. The initial model $\mathcal{M}_1$ is obtained following one of the training strategies presented in 3.2. At the k^{th} step of the CIL process, $k \in [\![2, K]\!]$, the classification model $\mathcal{M}_k$ recovers the weights of the model $\mathcal{M}_{k-1}$ obtained in step $k-1$ and is updated using the data subset $\mathcal{D}_k$ and the algorithm $Incr$. Many EFCIL algorithms [30] fine-tune all network weights at each incremental step, thus favoring plasticity. Alternatively, algorithms such as [22, 47] only retrain the classifier, thus favoring stability. As a compromise, it is also possible to freeze a part of the model and to update only the last layers. We cover these three cases in our experiments.

Scenario. A CIL scenario is characterized by the distribution of classes among the steps of the CIL process. We denote by b the proportion of the classes available in the initial step: $b = Card(\mathcal{C}_1)/Card(\mathcal{C})$. There are two commonly used scenarios [6] (i) equal splitting of classes across the steps or (ii) half of the classes in the first step and the rest of the classes are divided equally between subsequent steps.

3.2. Training strategies for the initial model

In the following, we describe the main characteristics of the training strategies used in our experimental study to obtain the initial model of the incremental learning process. Further experimental settings are reported in Section 4.

Network architecture. So far, most CIL methods have been proposed in combination with a convolutional neural network, but visual transformer (ViT) networks have recently gained popularity in CIL [13]. In order to provide a fair comparison between the two types of architecture, we use a ResNet50 [25], and a ViT-Small [11] network, which have a close number of parameters (23.5M and 22.1M parameters respectively).

Model initialization. At the first step of the CIL process, the weights of the model may either be randomly initialized or transferred from a pre-trained model. In the second case, depending on the choice of the user, the dataset $\mathcal{D}^*$ used for pre-training may either be an auxiliary dataset (e.g. ILSVRC [52]), referred to as *source* dataset, or the first data subset $\mathcal{D}_1$ of the incremental process.

Label availability. We consider that all examples from the target dataset $\mathcal{D}$ are labeled, and we experiment with both supervised learning and self-supervised learning to obtain the initial model using D_1. Labels may not be available for the external dataset $\mathcal{D}^*$. In this case, the training initialization is performed using a self-supervised pre-training algorithm (e.g., DINOv2 [43]).

4. Experimental setting

We describe the experimental parameters and the metrics we use to evaluate EFCIL models. The combination of parameters results in 1,248 experiments in total (Figure 1).

4.1. Initial training strategies

We compare different strategies for training an initial model, as summarized in Table 1. We use Resnet50 [25] and ViT-S [11] networks, which are representative of CNNs and transformers and have similar sizes. The training is done either using a self-supervised method (BYOL [20], DINOv2 [24], MoCov3 [9]) or a supervised one (DeiT, cross-entropy (CE)). We present results for pre-training with external data (i.e. ILSVRC [52] for BYOL, DeiT and CE; a 150M-images dataset + ILSVRC for DINOv2) and training on the first batch. We compare the effect of (i) freezing the weights of the pre-trained model or (ii) further optimizing the last layers of the model (e.g. the last convolutional block in ResNet50) on the initial data subset $\mathcal{D}_1$. The first type of experiment is denoted by the suffix "*-t*" (transfer), the second by the suffix "*-ft*" (fine-tuning). In the case where the pre-training algorithm is applied to $\mathcal{D}_1$ and not to $\mathcal{D}^\star$, there is no suffix.

4.2. Target datasets

For a comprehensive evaluation and to account for the diversity of visual tasks, we evaluate the training strategies on 16 target datasets, sampled from publicly available datasets. They cover different domains (plants, animals, landmarks, food, faces, traffic signs etc.), and different types of images (natural, drawings, paintings). IMN100_1 and IMN100_2 consist of 100 classes randomly selected from ImageNet-21k [10]. Flora is a thematic subset of ImageNet consisting of 100 classes belonging to the "flora" concept. IMN100_1, IMN100_2 and Flora have no mutual overlap and no overlap with ILSVRC [10, 52]. Amph100 and Fungi100, sampled from iNaturalist [60], respectively contain 100 classes of amphibians and fungi, selected so as to avoid overlap with animal and fungi classes from ILSVRC. We also sample 100-class subsets from other popular datasets: WikiArt100 [53], Casia100 [68], Food100 [7], Air100 [37], MTSD100 [36], Land100 [65], Logo100 [61] and Qdraw100 [21]. Finally, we consider three 1000-class subsets: Casia1k [68], Land1k [42], and iNat1k [60]. The number of training images per dataset varies from 60 to 750. More details on the datasets are provided in the supplementary material.

4.3. Incremental learning

EFCIL scenario b. We experiment on two widely used CIL scenarios [6, 27]. In the first scenario, the classes are equally distributed over 10 steps, e.g. 10 classes per step for a 100-class dataset. In the second scenario, half of the classes are learned in the initial step, and the other half is equally distributed over 10 incremental steps, e.g. $50+10\cdot5$ classes for a 100-class dataset.

CIL algorithm $Incr$. We experiment with one fine-tuning based algorithm, namely BSIL [30], which adds a balanced softmax without exemplars to LUCIR [27]. We also experiment with two fixed-representation-based algorithms, namely DSLDA [22] and FeTrIL [47].

4.4. Metrics

The performance of EFCIL models can be evaluated in several ways [38], discussed below.

Average incremental accuracy $\overline{Acc}$. In EFCIL, a model trained over a K-step incremental process is commonly evaluated using the average incremental accuracy [30, 69–71]. We denote it by $\overline{Acc}$ and compute it by:

$$\overline{Acc} = \frac{1}{K-1}\sum_{k=2}^{K} acc(\mathcal{M}_k, \bigcup_{i=1}^{k}\mathcal{D}_i) \quad (1)$$

where $acc(\mathcal{M}, D)$ is the accuracy of the model $\mathcal{M}$ on the dataset D. Following common practice in CIL [8, 47, 71], $\overline{Acc}$ does not take the accuracy of the initial model into account.

Average forgetting F. Average forgetting, denoted here by F, is computed by:

$$F = b \times f(\mathcal{D}_1) + \frac{1-b}{K-1}\sum_{k=2}^{K} f(\mathcal{D}_k) \quad (2)$$

where $f(\mathcal{D}_k) = \max_{k' \in [\![k,K]\!]} acc(\mathcal{M}_{k'}, \mathcal{D}_k) - acc(\mathcal{M}_K, \mathcal{D}_k))$ is the difference between the best performance achieved on the data subset $\mathcal{D}_k$ during the EFCIL process and the final performance of the model on this data subset [41].

Initial accuracy Acc_1. To unskew the statistical models we present in Section 5, we consider the initial accuracy, defined as the accuracy of the first model on the first data subset $\mathcal{D}_1$ and denoted by Acc_1, i.e. $Acc_1 = acc(\mathcal{M}_1, \mathcal{D}_1)$.

Final accuracy Acc_K. The accuracy of the last model of the incremental learning process on the complete dataset $\mathcal{D}$ is denoted by Acc_K, i.e. $Acc_K = acc(\mathcal{M}_K, \mathcal{D})$.

$\overline{Acc}$ gives more weight to past classes since at each step, the model is evaluated on all classes seen so far. Consequently, a high average incremental accuracy does not guarantee a high accuracy on the latest classes, particularly when half of the classes are learned initially. Forgetting is complementary to accuracy, as it focuses on model stability. A low value for F indicates that, on average, the performance for a given class remains stable over the incremental process.

5. Analysis of results

We present a statistical analysis of the results from Table 1, which highlights the effects of pre-training strategies and of EFCIL algorithms on EFCIL performance. The statistical model and associated findings are presented below.

5.1. Modeling causal effects

Our objective is to identify the primary factors that influence the performance of EFCIL algorithms. To interpret causal effects, we employ multiple linear regressions using the Ordinary Least Squares (OLS) method, following common statistical and econometric practices [4, 18]. In a linear regression, we aim to explain a target variable Y using explanatory variables X_i. The target variable is said endogenous, i.e. determined by its relationship with other variables. If the outcome of a variable X_i is selected by the experimenter, it is said to be exogenous, i.e. not caused by other variables. For a given experiment, we denote by Y the target metric accuracy (endogenous), $Data$ the evaluation dataset (exogenous), $Train$ the initial training strategy (exogenous), and $Incr$ the incremental algorithm (exogenous). We also consider the initial accuracy Acc_1 as an endogenous variable that may influence performance and can be controlled in our regressions. Other parameters, such as the total number of classes or the dataset, are examined as potential predictors of a metric.

An OLS regression fits a model of the following form:

$$Y = \beta_0 + \beta_1 Train + \beta_2 Incr + \beta_3 Data + \ldots + \varepsilon, \quad (3)$$

where the intercept β_0 is a scalar and ε is assumed to be normally distributed Gaussian noise. Since $Train$, $Incr$, and $Data$ are categorical, we encode them as one-hot vectors. Thus, β_1, β_2, and β_3 are vectors of the same size as the number of possible categories for each variable. To emphasize the explanatory variables and to simplify notation, in the following we denote the above regression model (Eq. 3) as "$Y \sim Train + Incr + Data + \ldots$".

Under appropriate assumptions[1], the estimated coefficients can be interpreted as estimated causal effects. The statistical significance of these effects is assessed by examining the *p-value* of the associated Student t-test for each coefficient [18]. Following established statistical practices [18], we set the significance value at .05. The significance, sign, magnitude, and interpretation of each estimated coefficient depend on the regression model. In particular, introducing more exogenous variables can cause instability in the regression. Therefore, for each metric Y, we adopt the following methodology to select only the most influential factors:

1. We use multiple regression models to represent the evaluation metric Y as a linear combination of different variables, or of the product of these variables. We ensure that the chosen regressions exhibit no collinearity or numerical issues[2].

[1] Primarily, non-perfect collinearity among exogenous variables and the normality of the estimated residuals $\hat{\varepsilon}$

[2] We assess this by examining the smallest eigenvalue of the Gram matrix of the data $X^T X$. Although Ridge or Lasso regression could address these concerns, their coefficients are less interpretable than those of OLS.

2. Subsequently, we select a regression model using the Akaike Information Criterion (AIC) [3], which regularizes the likelihood of the model based on its degrees of freedom.

3. We interpret the regression coefficients, the coefficient of determination R^2, and examine the Q-Q plot of the residuals $\hat{\varepsilon}$ to verify their normality.

4. Next, we conduct an Analysis of Variance (ANOVA) [18] on the regression to obtain aggregated statistics on the categorical variables.

5. Finally, we interpret the partial η^2 derived from the ANOVA as a measure of the importance of each variable.

A regression on a categorical variable requires the setting of a reference value for it. Therefore, the coefficient(s) associated with this categorical variable represent the causal effects of this variable *with respect to the reference level*. However, we want to compare all initial training strategies with each other to derive practical recommendations. Therefore, we use the following protocol to generate pairwise significant differences: (1) perform the same regression multiple times using a different reference category; (2) sum-up the pairwise comparisons in a double-entry matrix; (3) since we are performing multiple tests, we need to adjust the significance threshold of each test using Bonferroni correction [18], which consists of dividing the *p-value* threshold by the number of tests; (4) plot a heatmap of the pairwise comparisons between the choice of a parameter.

5.2. Metrics and confounding Factors

In Figure 2, we examine the relationship between the evaluation metrics defined in Subsection 4.4. We observe a strong positive correlation between $\overline{Acc}$ and Acc_K. There is a weak negative correlation between average incremental accuracy and forgetting, which is expected due to the inherent trade-off between stability (i.e. low forgetting) and plasticity in CIL (i.e. high performance on new classes). We note a significant correlation between average incremental accuracy and accuracy in the initial state. This correlation is expected since half of our experiments are done with half of the classes in the initial step. Additionally, the average incremental accuracy (Eq. 1) evaluates each model on each class, from the first occurrence of the class to the end of the incremental process, thus giving greater influence to earlier classes. Conversely, there is a weak correlation between forgetting and initial accuracy. This implies that the performance on the initial batch of classes does not significantly impact the model's stability throughout the incremental steps.

Based on these observations, we choose the average incremental accuracy $\overline{Acc}$ and the average forgetting F as the metrics of interest for our study, and include the effect of the initial accuracy in their models. Controlling the initial accuracy in a regression model is important to draw accurate conclusions: if pure accuracy is sought, then it can be

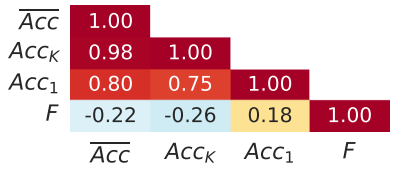

Figure 2. Correlation between the endogenous variables.

left out of the model. However, the goal of CIL algorithms is not solely to be accurate on average, but rather to be accurate while preventing forgetting. Hence, to analyse the actual incremental contribution of each method, initial accuracy should be included in the regression.

5.3. Factors influencing incremental performance

This subsection presents the aggregated influence of the considered parameters. The models and findings presented in Table 2 are obtained with the methodology presented in Subsection 5.1. More details on the obtained models can be found in the supplementary material.

Model	R^2	variable	η^2
$\overline{Acc} \sim Incr + Train + Data$	0.69	$Train$	0.32
		$Data$	0.24
		$Incr$	0.11
$\overline{Acc} \sim Acc_1 + Incr + Train + Data$	0.81	Acc_1	0.25
		$Incr$	0.22
		$Train$	0.10
		$Data$	0.06
$F \sim Incr + Train + Data$	0.71	$Incr$	0.61
		$Train$	0.06
		$Data$	0.03

Table 2. ANOVA results for each considered regression. Variables are significant at $p < 0.05$ and ordered by decreasing importance.

Main influences: In Table 2, the most significant factor affecting average incremental accuracy is the choice of initial training strategy. However, upon controlling the impact of initial accuracy, the selected incremental algorithm has a greater importance. This distinction is primarily attributed to BSIL, which exhibits an incremental accuracy 16 points below that of FeTrIL and DSLDA on average.

Regarding forgetting, the incremental algorithm is the most influential parameter. Here, this effect is not driven by any specific outlier method. Further analysis shows that initial accuracy also plays a significant role in predicting the level of forgetting. The associated regression coefficient is .16 ($\pm$.02), indicating that a 1-point increase in initial accuracy results in a 16-point increase of forgetting.

Given that accuracy ranges between 0 and 1, a lower initial accuracy decreases the likelihood of experiencing high levels of forgetting. Hence, a trade-off arises concerning the initial accuracy: while its enhancement greatly improves the average incremental accuracy, it also appears to amplify forgetting. This should be taken into account when comparing

CIL algorithms. From a research perspective, the incremental algorithm remains influential in the metrics, particularly when controlling for initial accuracy or focusing on forgetting. However, in practical applications of CIL, the final accuracy may be more important. Given its strong correlation with average incremental accuracy, increasing the initial accuracy becomes more advantageous in this case.

5.4. Comparison of initial training strategies

In Figure 3, we observe notable variations in accuracy among different initial training strategies, thus prompting the identification of three regimes:

1. **Strategies that surpass supervised learning without transfer:** MoCoV3-ft, DINOv2-t, BYOL-ft, SL(ResNet)-ft, MoCov3-t. These approaches exhibit superior performance by generating a robust latent space, whose features are transferable. MoCoV3-ft enhances its latent space by fine-tuning, enabling better generalization compared to other methods. DINOv2-t follows, leveraging its extensive self-supervised training on a very large amount of data. BYOL-ft and SL(ResNet)-ft closely follow, highlighting the advantage gained from additional adaptation steps on the target dataset following pre-training. MoCov3-t is fifth, showing that features generated through an adapted self-supervised method have a generalization capability that can be leveraged in CIL.

2. **Strategies that exhibit no significant improvement over supervised learning without transfer:** SL(ResNet)-ft, BYOL-t, SL(DeiT)-t. Our analysis underlines the capability of well-designed self-supervised methods to outperform supervised pre-training approaches.

3. **Strategies that underperform compared to supervised learning without transfer:** MoCoV3, BYOL, DINOv2-ft, SL(DeiT)-ft. The inferior performance of self-supervised methods can be attributed to the limited initial data. Furthermore, the challenging nature of fine-tuning for transformer models contributes to the underwhelming outcomes observed in these models.

The analysis of the average forgetting, illustrated in Figure 3 of the supplementary material, indicates that the majority of pairwise initial training strategies exhibit no significant distinctions. However, DINOv2-t exhibits lower forgetting compared to other strategies, including SL (ResNet). This is particularly remarkable considering that DINOv2-t has the highest initial accuracy. Conversely, fine-tuned transfer models (DINOv2-ft, SL(DeiT)-ft) also display a lower forgetting, albeit primarily attributed to their inherently low initial accuracy, which leaves little room for further decline in their accuracy.

5.5. Further analysis of initial training strategies

We now inquire whether the preceding general analysis can be nuanced in specific scenarios. To this end, we perform the same analysis as in the previous section by performing the regression on subsets of the data. All complementary graphs that justify the following statements can be found in the supplementary material.

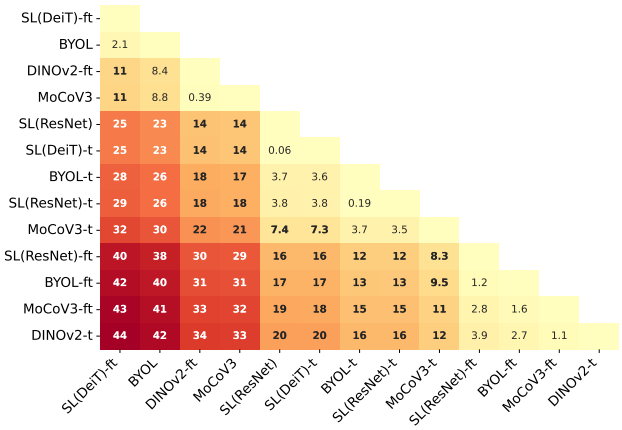

Figure 3. Accuracy gain by using strategy in row i over strategy in column j , e.g. "The accuracy of BYOL-ft is 17pts higher than SL(ResNet)". Only results in **bold** are statistically different.

Influence of the dataset. Regarding target datasets that are furthest from the pre-training dataset, the benefit of pre-training with or without fine-tuning is lower due to the domain gap. We note that specialized datasets, such as Qdraw100 and Casia100, also contain smaller images than those of ILSVRC. Whether the difference in performance is caused by a semantic gap or an image-size gap is unclear.

Influence of the incremental scenario. Regarding accuracy, we find that most differences among methods come from the scenarios with 50 initial classes or less. With 10 initial classes, all strategies that were previously not significantly better than SL(ResNet) start to outperform it. In scenarios with 50 initial classes, it becomes more difficult to precisely rank the top initial training strategies. In scenarios with 100 initial classes, no strategy is significantly better than any other one (which can come from the lower number of experiments with these scenarios).

Influence of Incremental method. We find that FeTrIL and DSLDA exhibit a similar pattern for $\overline{Acc}$ and F, contrary to BSIL. For FeTrIL and DSLDA, the differences between the best initial training strategies are less clear, but the general trend previously described still holds, in particular for the accuracy. The choice of the training strategy does not clearly impact the forgetting. On the other hand, BSIL is much more sensitive to the initial training strategy. Fine-tuned methods clearly outperform classical learning and plain transfer (except for DINOv2-t), whether it concerns the accuracy or the forgetting. Moreover, SL(ResNet) is a stronger baseline for BSIL than for the other methods when considering incremental accuracy.

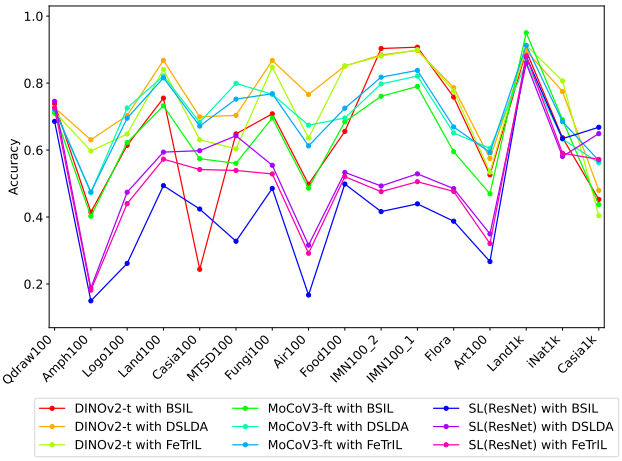

Figure 4. Interaction plot of the best strategies for different transfer types and for the 3 CIL algorithms. Similar slopes indicate similar behaviors. A change in slope indicates a change in behavior.

6. Discussion

We summarize our findings and propose recommendations for the design of EFCIL approaches.

Does the use of a model pre-trained on an external dataset $\mathcal{D}^\star$ always improve performance on the target dataset $\mathcal{D}$? Figure 4 highlights that no single initial training strategy outperforms the others on all datasets. As illustrated in Table 1, pre-training is clearly better on average, but there are exceptions. Intuitively, the use of a pre-trained model without fine-tuning (DINOv2-t in Figure 4), is clearly preferable for datasets such as IMN100_1 and Flora which are closely related to the dataset used for pre-training. Inversely, the supervised training method SL(ResNet) is better when the gap between the source and the target datasets is important, such is the case for Casia1k. MoCov3-ft is a good compromise since it leverages pre-training, but adapts the representation via partial fine-tuning. The initial training strategy should be selected by considering characteristics of the dataset such as: number of classes, number of samples per class, domain gap with pre-training, and size of the initial batch of classes.

In the absence of an external dataset, is it better to train the initial model in a supervised way or with a self-supervised learning method? As shown in Figure 3, supervised learning on the initial data is better on average. However, self-supervised learning is better when the amount of data available initially is limited, making it difficult to train a supervised model effectively.

Should the pre-trained model be fine-tuned on the first batch of data, or frozen? Existing EFCIL works that use pre-trained transformers keep their weights fixed [28, 46, 64]. This might be explained by the fact that fine-tuning these models might be detrimental in transfer learning [32]. Inversely, the performance of CNN-based training strategies, such as BYOL or MoCov3, increases after partial fine-tuning. This is explained by the fact that the layers of CNNs are reusable across tasks, while fine-tuning the last layers with initial target data improves transferability in subsequent EFCIL steps.

How does the performance of EFCIL algorithms vary with initial training strategies? Table 1 and Figure 4 show that the performance of BSIL varies much more than that of DSLDA and FeTrIL. This is particularly clear for transformer models, where BSIL performance is strongly degraded when fine-tuning of pre-trained models is used. In contrast, the variation of performance for DSLDA and FeTrIL is much lower when testing partial fine-tuning and transfer strategies on top of pre-trained models. This suggests that both initial training strategies are usable in practice for transfer-learning based EFCIL algorithms.

What is the impact of using transformers versus convolutional neural networks? The averaged results presented in Table 1 and the detailed ones from Figure 4 show that the difference between the best training strategies based on transformers and on CNNs is small. This is particularly the case when CNNs are pre-trained in a self-supervised manner and then partially fine-tuned on the initial batch of target data. Our finding echoes those reported in recent comparative studies of the two types of neural architectures which conclude that there is no absolute winner [48, 63]. The implication for EFCIL is that the use of both types of architecture should be explored in future works.

7. Conclusion

We perform an analysis of EFCIL in an evaluation setting that includes numerous and diverse classification tasks. We confirm the findings of existing comparative studies which have shown that no CIL algorithm is the best in all cases [6, 14, 38] and that algorithms based on transfer learning provide accuracy and stability for EFCIL [22, 28]. Our main finding is that the initial training strategy is the dominant factor influencing the average incremental accuracy, but that the choice of CIL algorithm is more important in preventing forgetting. Beyond the fact that there is no silver bullet approach to dealing with EFCIL, our in-depth statistical study quantifies the effect of different components of EFCIL approaches and thus enables informed decisions when designing new methods or implementing EFCIL in practice.

Acknowledgements. This work was supported by the European Commission under European Horizon 2020 Programme, grant number 951911 - AI4Media. It was made possible by the use of the FactoryIA supercomputer, financially supported by the Ile-de-France Regional Council.

References

[1] Samira Abnar, Mostafa Dehghani, Behnam Neyshabur, and Hanie Sedghi. Exploring the limits of large scale pre-training. In *International Conference on Learning Representations*, 2022. 2

[2] Touqeer Ahmad, Akshay Raj Dhamija, Steve Cruz, Ryan Rabinowitz, Chunchun Li, Mohsen Jafarzadeh, and Terrance E Boult. Few-shot class incremental learning leveraging self-supervised features. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3900–3910, 2022. 3

[3] Hirotogu Akaike. *Information Theory and an Extension of the Maximum Likelihood Principle*, pages 199–213. Springer New York, New York, NY, 1998. 6

[4] Joshua D Angrist and Jörn-Steffen Pischke. *Mostly harmless econometrics: An empiricist's companion*. Princeton university press, 2009. 5

[5] Eden Belouadah and Adrian Popescu. Deesil: Deep-shallow incremental learning. *TaskCV Workshop @ ECCV 2018.*, 2018. 3

[6] Eden Belouadah, Adrian Popescu, and Ioannis Kanellos. A comprehensive study of class incremental learning algorithms for visual tasks. *Neural Networks*, 135:38–54, 2021. 2, 3, 4, 8

[7] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101 – mining discriminative components with random forests. In *European Conference on Computer Vision*, 2014. 4

[8] Francisco M. Castro, Manuel J. Marín-Jiménez, Nicolás Guil, Cordelia Schmid, and Karteek Alahari. End-to-end incremental learning. In *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part XII*, pages 241–257, 2018. 5

[9] Xinlei Chen, Saining Xie, and Kaiming He. An empirical study of training self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9640–9649, 2021. 2, 4

[10] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Fei-Fei Li. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009), 20-25 June 2009, Miami, Florida, USA*, pages 248–255, 2009. 4

[11] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. 1, 4

[12] Arthur Douillard, Matthieu Cord, Charles Ollion, Thomas Robert, and Eduardo Valle. Podnet: Pooled outputs distillation for small-tasks incremental learning. In *Computer vision-ECCV 2020-16th European conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part XX*, volume 12365, pages 86–102. Springer, 2020. 3

[13] Arthur Douillard, Alexandre Ramé, Guillaume Couairon, and Matthieu Cord. Dytox: Transformers for continual learning with dynamic token expansion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9285–9295, June 2022. 4

[14] Eva Feillet, Grégoire Petit, Adrian Popescu, Marina Reyboz, and Céline Hudelot. Advisil - a class-incremental learning advisor. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 2400–2409, January 2023. 2, 8

[15] Enrico Fini, Victor G Turrisi Da Costa, Xavier Alameda-Pineda, Elisa Ricci, Karteek Alahari, and Julien Mairal. Self-supervised models are continual learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9621–9630, 2022. 3

[16] Robert M French. Catastrophic forgetting in connectionist networks. *Trends in cognitive sciences*, 3(4):128–135, 1999. 3

[17] Jhair Gallardo, Tyler L Hayes, and Christopher Kanan. Self-supervised training enhances online continual learning. In *British Machine Vision Conference (BMVC)*, 2020. 3

[18] James Gareth, Witten Daniela, Hastie Trevor, and Tibshirani Robert. *An introduction to statistical learning: with applications in R*. Spinger, 2013. 5, 6

[19] Robert Geirhos, Carlos R. M. Temme, Jonas Rauber, Heiko H. Schütt, Matthias Bethge, and Felix A. Wichmann. Generalisation in humans and deep neural networks. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. 2

[20] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in neural information processing systems*, 33:21271–21284, 2020. 2, 4

[21] David Ha and Douglas Eck. A neural representation of sketch drawings. *CoRR*, abs/1704.03477, 2017. 4

[22] Tyler L Hayes and Christopher Kanan. Lifelong machine learning with deep streaming linear discriminant analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 220–221, 2020. 1, 2, 3, 4, 5, 8

[23] Tyler L Hayes and Christopher Kanan. Online continual learning for embedded devices. In *Conference on Lifelong Learning Agents*, pages 744–766. PMLR, 2022. 1, 2, 4

[24] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738, 2020. 2, 4

[25] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Conference on Computer Vision and Pattern Recognition*, CVPR, 2016. 1, 4

[26] Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. Distilling the knowledge in a neural network. *CoRR*, abs/1503.02531, 2015. 2

[27] Saihui Hou, Xinyu Pan, Chen Change Loy, Zilei Wang, and Dahua Lin. Learning a unified classifier incrementally via rebalancing. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 831–839, 2019. 3, 4, 5

[28] Paul Janson, Wenxuan Zhang, Rahaf Aljundi, and Mohamed Elhoseiny. A simple baseline that questions the use of pretrained-models in continual learning. In *NeurIPS 2022 Workshop on Distribution Shifts: Connecting Methods and Applications*, 2022. 2, 3, 8

[29] Longlong Jing and Yingli Tian. Self-supervised visual feature learning with deep neural networks: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 43(11):4037–4058, 2020. 2

[30] Quentin Jodelet, Xin Liu, and Tsuyoshi Murata. Balanced softmax cross-entropy for incremental learning. In *Artificial Neural Networks and Machine Learning–ICANN 2021: 30th International Conference on Artificial Neural Networks, Bratislava, Slovakia, September 14–17, 2021, Proceedings, Part II*, pages 385–396. Springer, 2021. 1, 2, 3, 4, 5

[31] Simon Kornblith, Jonathon Shlens, and Quoc V. Le. Do better imagenet models transfer better? *CoRR*, abs/1805.08974, 2018. 2

[32] Ananya Kumar, Aditi Raghunathan, Robbie Jones, Tengyu Ma, and Percy Liang. Fine-tuning can distort pretrained features and underperform out-of-distribution. In *International Conference on Learning Representations*, 2022. 1, 2, 8

[33] Matthias De Lange, Rahaf Aljundi, Marc Masana, Sarah Parisot, Xu Jia, Ales Leonardis, Gregory G. Slabaugh, and Tinne Tuytelaars. Continual learning: A comparative study on how to defy forgetting in classification tasks. *CoRR*, abs/1909.08383, 2019. 3

[34] Zhizhong Li and Derek Hoiem. Learning without forgetting. In *European Conference on Computer Vision*, ECCV, 2016. 2, 3

[35] Divyam Madaan, Hongxu Yin, Wonmin Byeon, Jan Kautz, and Pavlo Molchanov. Heterogeneous continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15985–15995, 2023. 2

[36] Ahmed Madani and Rubiyah Yusof. Malaysian traffic sign dataset for traffic sign detection and recognition systems. *Journal of Telecommunication, Electronic and Computer Engineering (JTEC)*, 8(11):137–143, 2016. 4

[37] Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft. *arXiv preprint arXiv:1306.5151*, 2013. 4

[38] Marc Masana, Xialei Liu, Bartlomiej Twardowski, Mikel Menta, Andrew D. Bagdanov, and Joost van de Weijer. Class-incremental learning: survey and performance evaluation on image classification, 2021. 2, 3, 5, 8

[39] Michael Mccloskey and Neil J. Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. *The Psychology of Learning and Motivation*, 24:104–169, 1989. 3

[40] M Mermillod, A Bugaiska, and P Bonin. The stability-plasticity dilemma: investigating the continuum from catastrophic forgetting to age-limited learning effects. *Frontiers in Psychology*, 4:504–504, 2013. 2

[41] Seyed Iman Mirzadeh, Arslan Chaudhry, Dong Yin, Huiyi Hu, Razvan Pascanu, Dilan Gorur, and Mehrdad Farajtabar. Wide neural networks forget less catastrophically. In *International Conference on Machine Learning*, pages 15699–15717. PMLR, 2022. 5

[42] Hyeonwoo Noh, Andre Araujo, Jack Sim, Tobias Weyand, and Bohyung Han. Large-scale image retrieval with attentive deep local features. In *Proceedings of the IEEE international conference on computer vision*, pages 3456–3465, 2017. 4

[43] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023. 2, 4

[44] Oleksiy Ostapenko, Timothee Lesort, Pau Rodriguez, Md Rifat Arefin, Arthur Douillard, Irina Rish, and Laurent Charlin. Continual learning with foundation models: An empirical study of latent replay. In Sarath Chandar, Razvan Pascanu, and Doina Precup, editors, *Proceedings of The 1st Conference on Lifelong Learning Agents*, volume 199 of *Proceedings of Machine Learning Research*, pages 60–91. PMLR, 22–24 Aug 2022. 3

[45] German Ignacio Parisi, Ronald Kemker, Jose L. Part, Christopher Kanan, and Stefan Wermter. Continual lifelong learning with neural networks: A review. *Neural Networks*, 113, 2019. 3

[46] Francesco Pelosin. Simpler is better: off-the-shelf continual learning through pretrained backbones. *arXiv preprint arXiv:2205.01586*, 2022. 2, 3, 8

[47] Grégoire Petit, Adrian Popescu, Hugo Schindler, David Picard, and Bertrand Delezoide. Fetril: Feature translation for exemplar-free class-incremental learning. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 3911–3920, January 2023. 1, 2, 3, 4, 5

[48] Francesco Pinto, Philip HS Torr, and Puneet K. Dokania. An impartial take to the cnn vs transformer robustness contest. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XIII*, pages 466–480. Springer, 2022. 8

[49] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H. Lampert. icarl: Incremental classifier and representation learning. In *Conference on Computer Vision and Pattern Recognition*, CVPR, 2017. 2, 3

[50] Ricardo Ribani and Mauricio Marengoni. A survey of transfer learning for convolutional neural networks. In *2019 32nd SIBGRAPI Conference on Graphics, Patterns and Images Tutorials (SIBGRAPI-T)*, pages 47–57, 2019. 2

[51] Mark B Ring. Child: A first step towards continual learning. *Machine Learning*, 28(1):77–104, 1997. 2

[52] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael S. Bernstein, Alexander C.

Berg, and Fei-Fei Li. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015. 1, 4

[53] Babak Saleh and Ahmed Elgammal. Large-scale classification of fine-art paintings: Learning the right metric on the right feature. *arXiv preprint arXiv:1505.00855*, 2015. 4

[54] Youssef Tamaazousti, Hervé Le Borgne, Céline Hudelot, Mohamed El Amine Seddik, and Mohamed Tamaazousti. Learning more universal representations for transfer-learning. *CoRR*, abs/1712.09708, 2017. 2

[55] Chuanqi Tan, Fuchun Sun, Tao Kong, Wenchang Zhang, Chao Yang, and Chunfang Liu. A survey on deep transfer learning. In *International conference on artificial neural networks*, pages 270–279. Springer, 2018. 2

[56] Chi Ian Tang, Lorena Qendro, Dimitris Spathis, Fahim Kawsar, Cecilia Mascolo, and Akhil Mathur. Practical self-supervised continual learning with continual fine-tuning. *arXiv preprint arXiv:2303.17235*, 2023. 3

[57] Songsong Tian, Weijun Li, Xin Ning, Hang Ran, Hong Qin, and Prayag Tiwari. Continuous transfer of neural network representational similarity for incremental learning. *Neurocomputing*, 545:126300, 2023. 3

[58] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International conference on machine learning*, pages 10347–10357. PMLR, 2021. 2

[59] Gido M Van de Ven and Andreas S Tolias. Three scenarios for continual learning. *arXiv preprint arXiv:1904.07734*, 2019. 1

[60] Grant Van Horn, Oisin Mac Aodha, Yang Song, Yin Cui, Chen Sun, Alex Shepard, Hartwig Adam, Pietro Perona, and Serge Belongie. The inaturalist species classification and detection dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8769–8778, 2018. 4

[61] Jing Wang, Weiqing Min, Sujuan Hou, Shengnan Ma, Yuanjie Zheng, Haishuai Wang, and Shuqiang Jiang. Logo-2k+: A large-scale logo dataset for scalable logo classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34 (4), pages 6194–6201, 2020. 4

[62] Yu-Xiong Wang, Deva Ramanan, and Martial Hebert. Growing a brain: Fine-tuning by increasing model capacity. In *Conference on Computer Vision and Pattern Recognition*, CVPR, 2017. 3

[63] Zeyu Wang, Yutong Bai, Yuyin Zhou, and Cihang Xie. Can CNNs be more robust than transformers? In *The Eleventh International Conference on Learning Representations*, 2023. 8

[64] Zifeng Wang, Zizhao Zhang, Chen-Yu Lee, Han Zhang, Ruoxi Sun, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, and Tomas Pfister. Learning to prompt for continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 139–149, 2022. 2, 3, 8

[65] Tobias Weyand, Andre Araujo, Bingyi Cao, and Jack Sim. Google landmarks dataset v2-a large-scale benchmark for instance-level recognition and retrieval. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2575–2584, 2020. 4

[66] Guile Wu, Shaogang Gong, and Pan Li. Striking a balance between stability and plasticity for class-incremental learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1124–1133, 2021. 2

[67] Tz-Ying Wu, Gurumurthy Swaminathan, Zhizhong Li, Avinash Ravichandran, Nuno Vasconcelos, Rahul Bhotika, and Stefano Soatto. Class-incremental learning with strong pre-trained models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9601–9610, June 2022. 3

[68] Dong Yi, Zhen Lei, Shengcai Liao, and Stan Z Li. Learning face representation from scratch. *arXiv preprint arXiv:1411.7923*, 2014. 4

[69] Fei Zhu, Zhen Cheng, Xu-yao Zhang, and Cheng-lin Liu. Class-incremental learning via dual augmentation. *Advances in Neural Information Processing Systems*, 34, 2021. 2, 3, 5

[70] Fei Zhu, Xu-Yao Zhang, Chuang Wang, Fei Yin, and Cheng-Lin Liu. Prototype augmentation and self-supervision for incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5871–5880, 2021. 2, 3, 5

[71] Kai Zhu, Wei Zhai, Yang Cao, Jiebo Luo, and Zheng-Jun Zha. Self-sustaining representation expansion for non-exemplar class-incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9296–9305, 2022. 2, 3, 5